17

Supreme Court, U.S.
FILED
DEC 8 1995
CLERK

No. 94-2003

In the Supreme Court
OF THE
United States

OCTOBER TERM, 1995

LOTUS DEVELOPMENT CORPORATION,
Petitioner,
v.
BORLAND INTERNATIONAL INC.,
Respondent.

On Writ Of Certiorari To The United States Court Of Appeals For The First Circuit

Brief *Amicus Curiae* of Software Forum In Support of Respondent

DIANE MARIE O'MALLEY
Counsel of Record
MABEL NG
HANSON, BRIDGETT, MARCUS,
VLAHOS & RUDY
333 Market Street, Suite 2300
San Francisco, CA 94105-2173
(415) 777-3200
Attorneys for Amicus Curiae
SOFTWARE FORUM

December 1995

BEST AVAILABLE COPY

28 PP

TABLE OF CONTENTS

TABLE OF AUTHORITIES

CASES

	Page
Baker v. Selden, 101 U.S. 99, 25 L.Ed. 841 (1879)	19
Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141, 109 S.Ct. 971, 103 L.Ed.2d. 118 (1989)	8n
Campbell v. Acuff-Rose Music, Inc., ____ U.S.____, 114 S.Ct. 1164, 127 L.Ed.2d 500 (1994)	9
Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992)	10
Digital Equipment Corp. v. C. Itoh & Co., Ltd., 229 U.S.P.Q. 598 (D.N.J. 1985)	14n
Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991)	8
Fogerty v. Fantasy, Inc., ____U.S.____, 114 S. Ct. 1023, 127 L.Ed.2d 455 (1994)	6
Gates Rubber Co. v. Bando Chemical Industries, Ltd., 9 F.3d 823 (10th Cir. 1993)	14, 14n
Harper & Row Publishers, Inc. v. Nation Enters., 471 U.S. 539, 105 S. Ct. 2218, 85 L.Ed.2d 588 (1985)	8
Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F.Supp. 78 (D. Mass. 1992)	3n, 7, 10n
Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F.Supp. 203 (D. Mass. 1992)	3n
Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F.Supp. 202 (D. Mass. 1993)	3n
Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F.Supp. 223 (D. Mass. 1993)	3n, 10n

TABLE OF AUTHORITIES

CASES

	Page
Lotus Development Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995)	*Passim*
Lotus Dev. Corp. v. Paperback Software Int'l, 740 F.Supp. 37 (D. Mass. 1990)	2, 3, 11
NEC Corp. v. Intel Corp., 10 U.S.P.Q.2d 1177 (N.D. Cal. 1989)	12n
Sony Corp. of America v. Universal City Studios, Inc., 464 U.S. 417, 104 S.Ct. 774, 78 L.Ed.2d 574 (1984)	9
Xerox Corp. v. Apple Computer, Inc., 734 F.Supp. 1542 (N.D. Cal. 1990)	14

OTHER AUTHORITIES

P. Barnard, et al., *Learning and Remembering Interactive Commands,* 1982 Proc.: Hum. Factors in Computer Sys. 2	20n
K. Bury, et al., *Windowing vs. Scrolling on a Visual Display Terminal,* 1982 Proc.: Hum. Factors in Computer Sys. 41	20n
Computers & Intellectual Property: Hearings Before the Subcomm. on Courts, Intellectual Property, and the Admin. of Justice of the House Comm. on the Judiciary, 101st Cong., 2d Sess. 243 (1990)	4
Final Report of the National Commission on New Technological Uses of Copyright Works (1978)	15

TABLE OF AUTHORITIES

OTHER AUTHORITIES

Page

A. Hauptmann & B. Green, *A Comparison of Command, Menu-selection, and Natural Language Computer Programs,* 2 Behav. & Info. Tech. 163 (1983) 20n

D.S. Karjala, *"Copyright, Computer Software, and the New Protectionism,"* Jurismetrics J. 33, 59-60 (Fall 1987) 13n

G.K. Landy, "Macros and Broad Software Copyright Protection: The Latest Round in Lotus v. Borland," *The Computer Lawyer,* Vol. 11, No. 2 at 25 (Feb. 1994) 15n

J. Rosenberg, *Evaluating the Suggestiveness of Command Names,* 1982 Proc.: Hum. Factors in Computer Sys. 12 .... 20n

P. Samuelson & R.J. Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright "Look and Feel" Lawsuits,* 30 Jurismetrics J. 121 (Fall 1989) 17n

R. Stallman & S. Garfinkel, *Viewpoint: Against User Interface Copyright,* Comm. ACM, Nov. 1990, at 15 .... 19n

R.M. Whitmeyer, Comment, *A Plea for Due Processes: Defining the Proper Scope of Patent Protection for Computer Software,* 85 Nw. U. L. Rev. 1103 (1991) 8n

In the Supreme Court
OF THE
United States

OCTOBER TERM, 1995

No. 94-2003

LOTUS DEVELOPMENT CORPORATION,
Petitioner,

v.

BORLAND INTERNATIONAL INC.,
Respondent.

On Writ Of Certiorari To The United States Court Of Appeals For The First Circuit

BRIEF *AMICUS CURIAE*
SOFTWARE FORUM

INTEREST OF THE AMICUS CURIAE

Software Forum (the "Forum") respectfully submits this brief *amicus curiae* in support of affirmance of the decision of the United States Court of Appeals for the First Circuit below, *Lotus Development Corp. v. Borland Int'l, Inc.,* 49 F.3d 807 (1st Cir. 1995). The consent of the attorneys for both Respondent and Petitioner has been obtained.[1]

[1] Neither Lotus nor Borland is represented on the Forum's 20-member Board of Directors or its 15-member Advisory Board. The Forum, previously known as the Software Entrepreneurs' Forum, has filed *amicus* briefs in this matter with both the District Court and the Court of Appeals. This Brief was approved pursuant to the balloting procedures

The Forum, founded in 1983 and based in California's Silicon Valley, is a non-profit organization with a membership of more than 1,000 independent software developers, consultants and software providers. While the Forum is reluctant to burden this Court with another brief in this case, it believes it brings a valuable perspective to the issues pending before the Court. The Forum is the largest non-profit group in the United States that provides software entrepreneurs with resources needed to develop, market and sell software products. Its counsel are familiar with the briefs and questions presented. Its members are united by a desire to preserve their freedom to develop software products and businesses based on free-market principles and market-driven forces. Many of it members are precisely the "small developers" that the District Court and Lotus contended would be the "biggest losers of all" if Lotus's menu command hierarchy were not protected by copyright. *Lotus Dev. Corp. v. Paperback Software Int'l,* 740 F.Supp. 37, 75 (D. Mass. 1990) (*"Paperback"*), Pet. App. 183a, 249a. The Forum could not disagree more with the District Court's decision. Far from encouraging the creation of new software programs, the Forum views the position urged by Lotus and adopted by the District Court as leveling a potentially lethal blow to innovation and development in the software industry.

The Forum's members believe that there is additional argument needed on some important points which have not received sufficient attention. First, the Forum wishes to emphasize that the extension of copyright protection to computer menu command hierarchies inhibits the efforts of all — consumer and developer — to work in an organized environment. The broad copyright granted Lotus by the District Court frustrates this country's ability to compete

of the Forum which exercised complete control over its editorial content. Borland helped to defray the cost of printing.

and excel by foreclosing the use of standard items for the ultimate benefit and ease of use of all consumers (as was the case with Selden's System of double-entry booking which the Court of Appeals recognized as the now-almost universal T-accounts system). As aptly pointed out by the Court of Appeals, "under Lotus's theory, if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used." *Lotus Development Corp. v. Borland Int'l, Inc.,* 49 F.3d 807, 817-818 (1st Cir. 1995) ("Lotus"), Pet. App. 20a. The Forum does not believe that such relearning benefits the ultimate user or the industry as a whole and believes that it runs contrary to the purpose of the copyright laws to enrich the public through access to creative works.

Second, the freedom to develop software products is jeopardized if this Court fails to affirm the decision of the Court of Appeals. Indeed, the disturbing implications of the decisions of the District Court[2] and its reliance on *Paperback* so troubled the Forum's members that the Forum filed *amicus* briefs on the merits both in the District Court and the Court of Appeals. It does so here again with the hope of persuading this Court to preserve a fundamental characteristic of the software industry's innovation and development.

Ironically, in testimony before Congress immediately before this lawsuit was filed, Mitchell Kapor, the principal

[2] *Lotus Dev. Corp. v. Borland Int'l, Inc.,* 788 F.Supp. 78 (D. Mass. 1992) (*"Borland I"*), Appendix to the Petition for Certiorari ("Pet. App.") 145a; *Lotus Dev. Corp. v. Borland Int'l, Inc.,* 799 F.Supp. 203 (D. Mass. 1992) (*"Borland II"*), Pet. App. 106a; *Lotus Dev. Corp. v. Borland Int'l, Inc.,* 831 F.Supp. 202 (D. Mass. 1993) (*"Borland III"*), Pet. App. 71a; and *Lotus Dev. Corp. v. Borland Int'l, Inc.,* 831 F.Supp. 223 (D. Mass. 1993) (*"Borland IV"*), Pet. App. 29a.

developer of Lotus 1-2-3, captured what is at stake in this case:

> If you want to keep this industry as vibrant and successful as it's been, then a properly constructed intellectual property policy will respect protection but give preference to innovation. Over-protection of intellectual property is as pernicious as under-protection in its stifling effects on innovation and consequent loss to society.

Computers & Intellectual Property: Hearings Before the Subcomm. on Courts, Intellectual Property, and the Admin. of Justice of the House Comm. on the Judiciary, 101st Cong., 2d Sess. 243 (1990). To allow copyright protection of purely functional elements such as the 1-2-3 menu command hierarchy will have precisely such a "pernicious" effect on "innovation and consequent loss to society." The Forum will elaborate upon these points as succinctly as possible in the pages that follow.

SUMMARY OF ARGUMENT

This *amicus* brief makes two points. First, the Forum believes that in order to foster continuing innovation in the software industry, the scope of copyright protection of computer programs must be reasonably ascertainable and predictable. The uncertainty created by an *ad hoc* test necessarily deters new software development. The wider the scope of uncertainty, the greater the chilling effect. The resort to two modes of analysis in the proceedings below — the "fair use" doctrine discussed (but not endorsed) by Judge Boudin and the "literal/non-literal" analysis used by the District Court — necessarily leaves software developers with no means of assessing whether their conduct is permissible under the copyright laws.

Second, the Forum urges the Court to find menu command hierarchies to be uncopyrightable subject matter because of the value to public access obtained by allowing the most efficient means of performing a task to be available. Menu command hierarchies are not "computer programs" or "screen displays." Rather, they are an output of a program that in turn may be operated by the user to control the program. As the Court of Appeals correctly analogized, it is as though a video cassette recorder ("VCR") generated the "virtual" buttons that the user then employs to operate the VCR and view its screen. These VCR virtual buttons are not a proper subject of copyright.

The command hierarchy does not have an existence separate from the control function. A menu command hierarchy is the method by which a user operates a computer, just as surely as a handle is the method by which a user operates a hammer and a telephone keypad is the method by which a user operates a telephone. The Forum urges the Court to adopt a bright-line rule holding that such functional elements are unprotected under the copyright laws.

The Forum believes that the need for such clear demarcation is especially pressing in the software industry and that neither "fair use" nor "literal/non-literal" analysis provides the Forum members or other developers with a framework for making a reasonable evaluation of the scope of protection for software under the copyright laws. The District Court's decision also does not recognize the importance of the ultimate user's investment in learning the menu command hierarchy and what would be lost if the user had to learn a difficult menu command over and over and over each time he or she used a new program.

ARGUMENT

I.

THE NEED FOR A "BRIGHT LINE" TEST

This Court has recognized that the monopoly rights that Congress granted under the Copyright Act are limited in nature and have as their purpose "enriching the general public through access to creative works." *Fogerty v. Fantasy, Inc.*, ____ U.S. ____, 114 S. Ct. 1023, 1030, 127 L.Ed.2d 455, 465 (1994). To serve that purpose, this Court has held that "it is peculiarly important that the boundaries of copyright law be demarcated as clearly as possible." *Id.*

A. The Absence Of A Bright-Line Test Deters Software Innovation.

In the Forum's view, the need for clear rules identifying the boundaries of copyright protection, while always acute, is rendered even more so by three characteristics of the software industry. First, many if not most software developers are small entrepreneurs, not large corporations. These small firms have provided much of the force behind the successive waves of innovation that have propelled forward the information revolution. Such firms are extremely susceptible to the chilling effects of uncertain legal rules. Small entrepreneurs simply do not have the resources to support years of litigation in geographically-remote courts in order to get a product to the market. And even were such resources available, the cloud placed over a new product by lengthy litigation is potentially lethal not only to the product but also to the small developers' very existence.

In this respect, Borland's fate has provided developers with a cautionary note that has chilled the industry. When the District Court rendered its initial decision imposing an injunction on Borland, the developer community was incredulous. By the time Borland, which by the standards of most small developers is a *large* company, was able to get the injunction lifted, the company was driven out of the spreadsheet market. For Forum members, as well as for the industry, there is only one message — that even "unsuccessful" copyright litigation can crush smaller competitors. If the scope of copyright protection is in any way uncertain, prudent small software developers will simply avoid risks and curtail any new development.

A second characteristic of the software industry that adds to the need for bright-line rules is the short lifespan of most software products. Products placed under the cloud of litigation likely will be obsolete or unwanted by the time the product's developer ultimately prevails in litigation. Borland's experience again is illustrative. Two years after litigation began, Lotus still had not formulated "which elements of 1-2-3, separately or in combination, were copyrightable," much less which were copied. *Borland I,* 788 F.Supp. at 98, Pet. App. 180a. Unlike patent cases, where the entire prosecution and claim process is designed to put competitors and others on notice at the outset of the bounds of the claimed invention, a developer sued in a copyright case might still be guessing as to what was infringed more than half way through the life of a software program with a four-year life cycle. To steer clear of such traps, developers will avoid committing substantial resources to the development of new products if there is any uncertainty at all regarding the scope of copyright protection of some other software product.

Finally, the software industry needs clear guidelines because copyright issues in the field are likely to be recurring. Like most other technologies, software engineering advances incrementally. Software developers rarely construct programs that perform basic functions from scratch. Instead, they resort to software libraries or to the technical literature for efficient algorithms or modules of computer code.[3] In other words, software developers "build freely upon the ideas and information conveyed" by other programs. *Feist Publications, Inc. v. Rural Telephone Service Co.*, 499 U.S. 340, 349-350, 111 S.Ct. 1282, 1289-1290, 113 L.Ed.2d 358 (1991). Uncertainty regarding the extent to which such incremental advances are permitted under copyright law strikes at the very heart of the process by which software innovation takes place.[4]

B. "Fair Use" Analysis Would Be Undesirable As A Basis For Decision.

In his concurring opinion in the Court of Appeals, Judge Boudin observed that a different approach for decision in this case is an adaptation of the "fair use" doctrine. *Lotus,* 49 F.3d 807, 821; Pet. App. 27a-28a (Boudin, J., concurring). As Judge Boudin recognized, however, application of the fair use doctrine "would cause cost and delay," and "would also reduce the ability of the industry to predict outcomes." *Id.* at 821-822; Pet. App. 28a. Indeed, fair use requires *ad hoc* determinations that are the antitheses of the kind of bright-line rule that the software industry urgently needs.

The indeterminate nature of fair use analysis has been made clear by this Court. For example, in *Harper & Row Publishers, Inc. v. Nation Enters.,* 471 U.S. 539, 560, 105 S. Ct. 2218, 2230, 85 L.Ed.2d 588, 607 (1985), this Court said that because the doctrine is an equitable rule of reason, "no generally applicable definition is possible, and each case raising the question must be decided on its own facts." *Accord Sony Corp. of America v. Universal City Studios, Inc.,* 464 U.S. 417, 448 n. 31, 104 S.Ct. 774, 792 n. 31, 78 L.Ed.2d 574, 596 n. 31 (1984). Similarly, in *Campbell v. Acuff-Rose Music, Inc.,* ____ U.S. ____, 114 S.Ct. 1164, 1170-1171, 127 L.Ed.2d 500, 514-515 (1994), this Court emphasized that fair use "calls for case-by-case analysis" and that the four statutory factors listed in the Copyright Act are not to be treated in isolation, but rather "[a]ll are to be explored, and the results weighed together, in light of the purposes of copyright."

Consideration of the fair use factors serves to highlight further the lack of correspondence between the issues in this case and the concerns to which fair use is directed. For example, under the first factor, Judge Boudin recognized that Borland's commercial use could weigh against a finding of fair use. *Lotus,* 49 F.3d at 821, Pet. App. 27a-28a. Yet, the commercial nature of Borland's use in which Borland was "merely trying to give former Lotus users an option to exploit their own prior investment in learning or in macros" is at the heart of the conduct that Judge Boudin thought should be protected. *Id.,* Pet. App. 27a. Similarly, the fourth factor weighs against a finding of fair use if the defendant's conduct "would result in a substantially adverse impact on the potential market" for the original. *Campbell,* 114 S.Ct. at 1177. It is precisely in such instances of potential market impact, however, that Judge Boudin believed that the defendant's conduct should be privileged. *Lotus,* 49 F.3d at 822, Pet. App. 28a. ("Indeed, to the extent that Lotus' menu is an important standard in the industry, it might be argued that any use ought to be deemed privileged.").

Attempting to resolve this case by adapting fair use analysis is thus an effort to force a square peg into a round hole. The effort is likely to lead to even greater uncertainty

[3] *See, e.g.,* R.M. Whitmeyer, Comment, *A Plea for Due Processes: Defining the Proper Scope of Patent Protection for Computer Software,* 85 Nw. U. L. Rev. 1103, 1110 (1991).

[4] The Court's patent cases recognize this incremental process as lying at the heart of technological innovation. *See, e.g., Bonito Boats, Inc. v. Thunder Craft Boats, Inc.,* 489 U.S. 141, 156-157, 109 S.Ct. 971, 980-981, 103 L.Ed.2d. 118, 137-138 (1989).

than is already inherent in fair use analysis. For software developers, therefore, fair use analysis is a singularly ill-suited vehicle for addressing the issues posed by this case.

C. "Literal/Non-literal" Analysis Is Similarly Problematic.

In expressly eschewing the application of bright-line rules,[5] the District Court took the position that the organization, structure and labels attached to the 1-2-3 menu hierarchy were "non-literal" elements of a computer program and that these "non-literal" elements, together with a program's "literal" elements, are copyrightable.[6] As developers, the Forum is concerned about its members' ability to predict whether a particular element will be protected under this literal/non-literal analysis. In this respect, at least, the Forum concurs with the (somewhat understated) assessment of the Court of Appeals in *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 712 (2d Cir. 1992), that "[t]o be frank, the exact contours of copyright protection for non-literal program structure are not completely clear."

The District Court's decision amply illustrates the *ad hoc* and unpredictable nature of determinations made under a literal/non-literal analysis. For example, the District Court concluded that Borland was a copier, at best, and a pirate, at worst, despite the facts that (1) Borland was not found to have copied, paraphrased, translated or used any of Lotus's code; (2) Lotus did not contend (and, so far as the Forum has been able to ascertain, indeed has tacitly conceded) that Borland's code did not use the same structure or organization as Lotus's code; and (3) Borland's product "looks substantially different" in terms of screen display, and hence did not infringe 1-2-3's pictorial displays. Borland, in fact, took far less of 1-2-3, from both a qualitative and quantitative point of view, than 1-2-3 took from such predecessor products as VisiCalc, VisiTrend/Plot, Multiplan, and Prokey. Yet, the District Court here, applying the literal/non-literal test, concluded that Borland's use of Lotus's command words was actionable, while it found in the earlier *Paperback* decision that Lotus's conduct in taking the spreadsheet metaphor, the use and identification of various function keys, the two-line moving cursor, the use of highlighting, and even the VisiTrend Plot command words themselves, was not only permissible but even worthy of approbation. *See Paperback*, 740 F.Supp. at 63-67, Pet. App. 227a-234a.

From the perspective of developers, these respective outcomes are frankly inexplicable and further complicate the already confusing copyright issues facing small developers. The *reason* the District Court reached such inexplicable outcomes, however, is easy to understand: as the Court of Appeals pointed out, the District Court's test asked the wrong question. *Lotus*, 49 F.3d at 815, Pet. App. 14a-15a. The test applied by the District Court in *Paperback* sought to distinguish between "the idea and its expression," *Paperback*, 740 F.Supp. at 65, Pet. App. 229a; but that is not the appropriate question to ask in determining the copyrightability of a functional work. To decide whether a wall socket is copyrightable, the question is not whether there is more than one way to express the "idea" of a wall socket or whether someone has used someone else's "expression" of the concept of a wall socket. Instead, with wall sockets — as with menu command hierarchies — the appropriate question is the one posed by the Court of Appeals: is the item functional and therefore foreclosed from protection by Section 102?

D. Extending Copyright Protection To Functional Elements Such As Menu Command Hierarchies

[5] *Borland I*, 788 F.Supp. at 92, Pet. App. 169a.

[6] *Borland IV*, 831 F.Supp. at 232-233, Pet. App. 43a-44a.

Could Undermine The Software Industry As A Whole.

It is significant that no software developers have expressed support for the position urged by Lotus. Indeed, support for the copyright scheme proposed by Lotus comes from an industry which has in past litigation unsuccessfully attempted to use copyright to prevent competition in the marketplace. Had the industry been successful in that litigation, it would have gained a monopoly position otherwise only available under patent law. Furthermore, that monopoly position would have lasted 75 years rather than the significantly shorter term available for patents. In addition, the safeguards in effect under patent law, specifically the requirements to disclose the invention, to present specific claims, and to show non-obviousness and novelty, would be completely evaded.

Illustrative is Intel's copyright litigation against rival NEC (and, three years later, against AMD). Intel long has enjoyed market dominance as the maker of the chips at the heart of all IBM-compatible personal computers. NEC, however, engaged in a full-scale effort to develop a microprocessor that would be compatible with Intel's but that would contain improved performance. Intel sued for copyright infringement on the ground that NEC had copied its "microcode," which is the set of roughly ninety machine-language instructions that tell the microprocessor which transistors to actuate. Ultimately, Intel's suit was unsuccessful as the District Court in that case found no infringement.[7] As one commentator summarized, what was at stake in the litigation, and in Intel's subsequent suit against AMD, was its ability to monopolize the microprocessor market indefinitely:

> Given that writing the actual microcode is a relatively small part of the cost of overall chip development, it seems almost incredible that a company in [NEC's] position would copy microcode after independently developing everything else, unless they thought copying was necessary for compatibility or was dictated by the hardware environment. Even assuming copying, however, to find copyright infringement is to give to this most fundamental form of computer software technology a level of copyright protection never before afforded to a technological product. . . . [T]he defendant can wholly enjoin the sale of the plaintiff's chip, thereby depriving the public of the advantages of a technologically superior product. Moreover, this position will be retained for seventy-five years, or at least until the original chip becomes obsolete through later technological developments. This seems a very high price for the public to pay to remedy a "wrong" that would not be at all wrong or unfair if it involved any type of technology other than one that effects machine operation through manipulation of human-understandable symbols.[8]

Xerox has been no less ambitious in its effort to use the copyright laws. In the late 1970's, Xerox, the inventor of the "mouse," developed a mouse-driven computer that "alleg-

[7] *See NEC Corp. v. Intel Corp.*, 10 U.S.P.Q.2d 1177 (N.D. Cal. 1989).

[8] D.S. Karjala, *"Copyright, Computer Software, and the New Protectionism,"* Jurismetrics J., at 59-60 (Fall 1987).

edly was the first to introduce fanciful visual displays and graphical images to aid user interaction with the computer." *Xerox Corp. v. Apple Computer, Inc.*, 734 F.Supp. 1542, 1544 (N.D. Cal. 1990). Xerox sued Apple, claiming that the copyright registration of Xerox's workstation included its computer program and screen displays. Although the suit was unsuccessful, Xerox had been remarkably forthcoming about the breadth of the monopoly that it sought. According to Xerox, by virtue of its copyright registration, it had "the right to control the future of graphic user interfaces in the 1990's and beyond."[9] *Id.* at 1545. Graphical user interfaces now are central to much of personal computing. Such a monopoly by Xerox, and the resulting lack of competition, would have significantly hindered the further development and refinement of that extremely important advance in the usability of software.

DEC and Gates Rubber each have sought specifically to use copyright protection of menu displays in litigation against their hardware competitors.[10] The example of *Gates Rubber* is particularly notable inasmuch as it shows that Lotus's proposed expansion of the copyright laws would be likely to have consequences well beyond the computer industry. Gates Rubber is the dominant manufacturer of rubber belts for use in industrial machinery and has developed a software program for performing the complicated calculations needed to determine the proper belt for a particular machine. It sued its competitor on the ground that the competitor's introduction of a rival software program violated Gates's copyright in a variety of functional

[9] The District Court responded to this assertion with the observation that "courts may on occasion break up monopolies, but should not create them." *Id.* at 1545, n.5.

[10] *See Gates Rubber Co. v. Bando Chemical Industries, Ltd.*, 9 F.3d 823, 843-844 (10th Cir. 1993); *Digital Equipment Corp. v. C. Itoh & Co., Ltd.*, 229 U.S.P.Q. 598, 603 (D.N.J. 1985).

program outputs, including its menus and sorting criteria for the entry of data. *Gates Rubber*, 9 F.3d at 843-844. As the use of software spreads even to such typically industrial uses, it seems clear that Lotus's claim, described by one commentator as a "broadening of copyright protection for program command protocols without logical limit,"[11] ultimately would touch virtually every aspect of American business.

Finally, and closer to home, it is not only other hardware makers that likely would be the targets of such copyright claims. In its Final Report, CONTU noted that the emergence of a separate software industry was tied to the inability of hardware manufacturers to control entry into the software market. Citing to a study commissioned by it, CONTU observed:

> Whatever their historical dominance, the hardware corporations lack the ability to control entry into the software market, and . . . their market shares are being steadily eroded by the independents. Thus, we can tentatively conclude that protection of software . . . serves to benefit consumers by enhancing competition and increasing long-run supply.

M. Breslow, A. Ferguson and L. Haverkamp, *An Analysis of Computer and Photocopying Issues From the Point of View of the General Public and the Ultimate Consumer* at IV-13 (1978), *quoted in Final Report of the National Commission on New Technological Uses of Copyright Works* at 24 (1978).

This inability to "control entry," however, assumes that hardware manufacturers cannot obtain a copyright in the

[11] G.K. Landy, "Macros and Broad Software Copyright Protection: The Latest Round in Lotus v. Borland," *The Computer Lawyer*, Vol. 11, No. 2, at 25 (Feb. 1994).

specifications needed to operate their machines. Yet, that is *precisely* what Lotus seeks to claim. Lotus 1-2-3 serves as a platform for the operation of macros in exactly the same way that computer hardware serves as a platform for the operation of applications software. Obviously, Lotus's parent, IBM, and other computer makers would benefit enormously from the ability to monopolize the control-point to their machines and, accordingly, recapture their "historical dominance" in the production of software in this country. The losers, on the other hand, would be not only the software industry, but also consumers and the American economy as a whole.

II.

MENU COMMAND HIERARCHIES ARE FUNCTIONAL AND HENCE NOT COPYRIGHTABLE.

As the authors of thousands of software programs, the Forum members support copyright protection of computer programs. It is fitting that copyright law protects against a "software pirate's" unauthorized copying of diskettes, prohibits users from unauthorized copying, and prevents copying of source code directly or indirectly by translating that code into another programming language. Such protection against the copying of source code and object code is appropriate. Moreover, because the statute protects audiovisual and pictorial works, it is also appropriate that the screen displays generated by a computer program are protected.

A menu command hierarchy, however, is neither a pictorial work nor a computer program. Although the way that the command hierarchy is *displayed* conceivably could be found to be sufficiently expressive to warrant protection, Lotus has made clear in its brief that it does not claim that Borland copied "the color, style, layout, or format in which the words or menus are displayed on the screens." Brief for the Petitioner, p. 6.

At the same time — and a point that Lotus fails to address in a meaningful way — the menu command hierarchy is not protectable as a "computer program." The menu command hierarchy is an *output* of the program, not the program itself. In defining "computer program," Section 101 provides that a program "is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." Pet. App. 272a. Although the menu command hierarchy may be used to generate such a "set of statements or instructions" — a kind of program known as a "macro" — it does not provide such an operational role itself. Put differently, it is the keyboard, not the composition.

A software developer can duplicate a menu command hierarchy quite closely without access to any of its competitor's underlying computer instructions, and Lotus apparently concedes that Borland did so here. Indeed, the Lotus code was never even entered into evidence in this case. Trial Tr. of Apr. 1, 1993, p. 5-136 to 5-137, Joint Appendix at 300. Consequently, this is a case where the copying of computer code or the structure, sequence or organization of the program is not at issue.[12] Instead, at issue is whether a program output is protected apart from the protection provided to pictorial and audiovisual works. The Court of

[12] Inasmuch as legal protection for computer programs is directed principally at providing an incentive for the creation of new programs, it is noteworthy that at least one survey has found the developer community to be hostile to extending intellectual property protection even to the structure or organization of computer code. According to the survey, an overwhelming majority supported intellectual property protection for source code (93%) and object code (85%), but they "strongly rejected the idea of protecting the 'structure, sequence, and organization' of computer programs." P. Samuelson & R.J. Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright "Look and Feel" Lawsuits*, 30 Jurismetrics J. 121, 129, 135 (Fall 1989).

Appeals correctly determined that it is not. Considered solely as output, the command hierarchy is an uncopyrightable "system." In context, however, there is a further reason why the command hierarchy is not protected: as the Court of Appeals found, the command hierarchy provides the program's "method of operation." Its role is no different than that played by pedals on a bicycle; and it is no more subject to protection under the copyright laws than are bicycle pedals.

A. Even Standing Alone, The Command Hierarchy Is An Unprotected "System."

Even considered apart from its functional role in operating a computer program, a menu command hierarchy is a non-copyrightable "system" under 17 U.S.C. Section 102(b). To illustrate, imagine a young Thomas Jefferson writing a program that creates a "virtual" Library of Congress. In running Mr. Jefferson's program, the user initially faces a screen depiction of the Library of Congress, which one can enter by clicking on the Library's front door. The user then sees a screen showing a large rotunda, with three wings marked "[ ]," "[ ]," and "Memory." Clicking on "Memory" calls up a screen showing a hallway divided between "Natural History" and "Human History." Clicking on "Human History" in turn reveals rooms labeled by country, and clicking the door for "U.S. History" reveals a room with many stacks. The stack marked "Civil War" has a shelf designated "Abraham Lincoln," and that shelf has on it the Gettysburg Address.

In considering the scope of copyright protection for this work, one can readily identify the source code and object code written by Mr. Jefferson as a protected computer program. The screen displays are pictorial or audiovisual works, and hence are protected as well. The Gettysburg Address, although generated in this instance through operation of the computer program, is in the public domain, and

hence is no more subject to copyright protection than are facts or ideas. Finally, the hierarchy by which the "books" are organized would not appear to be copyrightable, any more than such a system would be if devised wholly apart from the use of computers. To hold otherwise would seem to open up a radically expanded view of the scope of copyright, by which everything from the Dewey Decimal System to a system for arranging bookkeeping columns and headings could be subject to copyright — a result fundamentally at odds with *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879).

As the Court of Appeals recognized, the menu command hierarchy is also unprotected for another reason. The menu command hierarchy is not simply a system, but the means by which users can operate their computer program. This final point is addressed in the next section.

B. The Menu Command Hierarchy Is An Unprotected Method Of Operation.

As noted above, most if not all machines have what may be termed a "user interface." A user interface "is what the user must learn in order to operate a machine. The user interface of a typewriter is the layout of the keys. The user interface of a car includes a steering wheel for turning, pedals to accelerate and reduce speed, a lever to signal turns, etc."[13] In the case of a computer program, the user interface typically consists of the display on a screen of textual or graphic elements (icons) that enable the user to direct the operation of the program.

Computer programs have become increasingly universal, and that trend will only continue. For example, keyboards on computers, numeric pads on ATM machines and tele-

[13] R. Stallman & S. Garfinkel, *Viewpoint: Against User Interface Copyright*, Comm. ACM, Nov. 1990, at 15.

phones, and buttons on VCRs no longer operate mechanically, but instead operate through the use of software. Moreover, it seems likely that vocal commands will soon replace screen displays in many contexts. In such instances, there would be no visual depiction of the command hierarchy and no words attached to the hierarchy other than those spoken by the users themselves.

In all of these examples, what developers strive to maximize to the benefit of the ultimate consumer is not the "expressiveness" of the user interface, but its functionality. Indeed, an entire field of study known as "human factor analysis" is devoted to the empirical analysis of interface issues for purposes of optimizing efficiency and effectiveness. Thus, for example, researchers have examined choices among various abbreviations and command names,[14] the relative effectiveness of command-based, menu-driven, and natural language programs,[15] and the choice of windowing or scrolling.[16]

In his testimony in this case, Mitchell Kapor confirmed that he (like other developers) strived to devise a user interface with maximum efficiency and effectiveness. Thus, for example, menu command words were chosen to "intelligently convey to the user the purpose of each command and its underlying functionality." Affidavit of Mitchell D. Kapor ¶77, Pet. App. 291a. Kapor sought to achieve an organization that would "reflect a structured approach that communicated the product's underlying functionality." In other words, Kapor followed sound principles of product engineering in constructing the user interface.

[14] *See, e.g.*, J. Rosenberg, *Evaluating the Suggestiveness of Command Names*, 1982 Proc.: Hum. Factors in Computer Sys. 12; P. Barnard, et al., *Learning and Remembering Interactive Commands*, 1982 Proc.: Hum. Factors in Computer Sys. 2.

[15] *See, e.g.*, A. Hauptmann & B. Green, *A Comparison of Command, Menu-selection, and Natural Language Computer Programs*, 2 Behav. & Info. Tech. 163 (1983).

[16] *See, e.g.*, K. Bury, et al., *Windowing vs. Scrolling on a Visual Display Terminal*, 1982 Proc.: Hum. Factors in Computer Sys. 41.

Certainly a well-engineered user interface entails considerable creativity and inventiveness. That is far different than the District Court's apparent holding that, especially in a feature of industrial design as to which there is theoretically an optimally efficient answer, one developer can bar others from building upon its efforts — without ever having shown that its own effort constituted a material advancement over the previous efforts of others. Lotus has never shown that its menu command hierarchy constituted a new and nonobvious improvement in the development of user interfaces. As the Court of Appeals found, Borland therefore should be free to build upon it. The Court explained, in reasoning with which the Forum concurs entirely:

> [I]n most contexts, there is no need to "build" upon other people's expression, for the ideas conveyed by that expression can be conveyed by someone else without copying the first author's expression. In the context of methods of operation, however, "building" requires the use of the precise method of operation already employed; otherwise, "building" would require dismantling, too. Original developers are not the only people entitled to build on the methods of operation they create; anyone can. Thus, Borland may build on the method of operation that Lotus designed and may use the Lotus menu command hierarchy in doing so.

Lotus, 49 F.3d at 818, Pet. App. 21a.

If this Court were to reverse the Court of Appeals, it will substantially frustrate the efforts of the software industry to create software for the ultimate consumer in an organized and efficient manner. The Forum urges the Court to adopt a bright-line rule creating a "safe harbor" in which software developers are free to utilize the functional elements of the output of computer programs, including the program's menu command hierarchy. Such a rule will provide developers like the members of the Forum with needed protection against the copying of program code or the artistic elements of screen displays, while allowing the pace of innovation in the software industry to continue unimpeded.

CONCLUSION

The judgment of the Court of Appeals for the First Circuit should be affirmed.

Respectfully submitted,

DIANE MARIE O'MALLEY
Counsel of Record
MABEL NG
HANSON, BRIDGETT, MARCUS,
VLAHOS & RUDY
333 Market Street, Suite 2300
San Francisco, CA 94105-2173
(415) 777-3200
Attorneys for Amicus Curiae
SOFTWARE FORUM

December 1995